

### Amendments to the Claims

Please add new claims 28 and 29 as indicated below. For Examiner convenience, all claims are listed below. This listing of claims will replace all prior versions, and listings, of claims in the application:

1           1.     (Original)     An application programming interface (API) for enhancing  
2 data network communication, comprising:  
3           an identify address function including programming instructions for identifying a  
4 stored node address stored by a base driver for a network interface associated with the  
5 base driver; and  
6           an update node address function including programming instructions for directing  
7 the base driver to update the stored node address with a new node address in a  
8 configuration storage of the base driver, and in a receive address filtering table for the  
9 network interface.

10  
11           2.     (Original)     The API of claim 1, wherein the identify address function  
12 includes submitting a request to the base driver, to which is received a response  
13 including the node address stored by the base driver.

14  
15           3.     (Original)     The API of claim 1, wherein the identify address function  
16 includes programming instructions for inspecting the configuration storage of the base  
17 driver, such storage having an entry identifying the stored node address.

1           4.     (Previously Presented)     An API according to claim 1, further  
2 comprising:

3           a driver identification function including programming instructions for sending an  
4 identity-check request to the base driver, said base driver providing a response selected  
5 from a group consisting of: a predetermined identifier, a base driver revision number,  
6 and an identification of a vendor of the base driver.  
7

8           5.     (Original)     An API according to claim 4, wherein the predetermined  
9 identifier is a copyright string for the vendor of the base driver.  
10

11          6.     (Original)     An article of manufacture, comprising a computer readable  
12 medium having encoded thereon programming instructions capable of directing a  
13 processor to perform operations of:

14           an identify address function for identifying a stored node address stored by a  
15 base driver for a network interface associated with the base driver; and

16           an update node address function for directing the base driver to update the  
17 stored node address with a new node address in a configuration storage of the base  
18 driver, and in a receive address filtering table for the network interface.  
19

20          7.     (Original)     An API according to claim 1, further comprising:

21           a first transmission function including programming instructions for re-transmitting  
22 data, received in a compatible format from a network source, in an incompatible format  
23 to a network destination; and

1 a second transmission function including programming instructions for re-  
2 transmitting data, received in the incompatible format from the network destination, in  
3 the compatible format to the network source.

4  
5 8. (Original) An API according to claim 7, further comprising:  
6 a report capabilities function including programming instructions for sending the  
7 base driver a request to have the base driver report its capabilities;  
8 a receive capabilities function including programming instructions for receiving a  
9 response including said capabilities;  
10 wherein the incompatible format is formatted according to said capabilities.

11  
12 9. (Original) An API according to claim 7, further comprising:  
13 a virtual LAN function including programming instructions to direct the base driver  
14 to enter a desired virtual LAN operative state; and  
15 a disconnect function including programming instructions to notify the base driver  
16 that the API has concluded communications with the base driver.

17  
18 10. (Canceled)

19  
20 11. (Original) An API according to claim 1 for providing transparent fail-  
21 over from a first network interface to a second network interface, further comprising:  
22 a status function including programming instructions for polling a first base driver  
23 for the first network interface to detect a failure of said first network interface;

1 wherein the update node address function includes a function to direct a second  
2 base driver for the second network interface to store the node address of the first  
3 network interface as the stored node address for the second base driver.

4  
5 12. (Original) An API according to claim 11, in which a Novell ODI  
6 compliant network is utilized for network communication, and wherein the update node  
7 address function uses at least one ODI MLID Control Routine.

8  
9 13. (Previously Presented) An article of manufacture, comprising a  
10 computer readable medium having encoded thereon instructions to direct a processor to  
11 perform an API having:

12 an identify address function for identifying a stored node address stored by a  
13 base driver for a network interface associated with the base driver;

14 an update node address function for directing the base driver to update the  
15 stored node address with a new node address;

16 a status function in communication with a first base driver for the first network  
17 interface to detect a failure of the first network interface; and

18 a failover function to direct a second base driver for the second network interface  
19 to store the node address of the first network interface as the stored node address for  
20 the second base driver, and to store the node address of the first network interface in a  
21 receive address filtering table for the second network interface.

22

1           14.   (Original)   An API according to claim 1 for providing transparent load  
2 balancing of data transmissions directed towards the network interface by distributing  
3 such data across a second network interface, further comprising:

4           a queue monitoring function including programming instructions for detecting a  
5 workload of the first network interface; and

6           a distribution function including programming instructions for routing a portion of  
7 said data transmissions through the second network interface, said distribution function  
8 utilizing the update node address function to associate the node identifier of the first  
9 network interface with the second network interface.

10  
11           15.   (Previously Presented)   A networking method, comprising:

12           receiving first network traffic with a protocol stack;

13           sending said first traffic to an intermediary layer;

14           routing said first traffic to a virtual interface driver;

15           repackaging said first traffic by the virtual interface driver, and providing said  
16 repackaged traffic to a virtual protocol stack;

17           sending said repackaged traffic to the intermediary layer;

18           routing said repackaged traffic by the intermediary layer to an interface driver for  
19 a network interface having a node address memory;

20           identifying a failed network interface having a node address; and

21           storing the node address in the node address memory.

1           16.   (Previously Presented) A method according to claim 15, further  
2 comprising:  
3           routing network traffic for the failed network interface through the fail over  
4 network interface.

5  
6           17.   (Original)   A method according to claim 16, further comprising:  
7           wherein said first network traffic is received in a first protocol format, and said  
8 repackaged traffic is in a second network protocol format different from the first protocol  
9 format.

10  
11           18.   (Previously Presented) A method according to claim 16, wherein locating  
12 the fail over network interface comprises:

13           submitting a node identification request to a base driver for a potential fail over  
14 network interface; and

15           receiving a response from said driver, said response including an authentication  
16 string;

17           verifying said authentication string has a predetermined value before said  
18 potential fail over network interface is used as the fail over network interface.

19  
20           19.   (Previously Presented)   An article of manufacture, comprising a  
21 computer readable medium having encoded thereon instructions to direct a processor to  
22 perform the operations of:

23           receiving first network traffic with a protocol stack;

1            sending said first traffic to an intermediary layer;  
2            routing said first traffic to a virtual interface driver;  
3            repackaging said first traffic by the virtual interface driver, and providing said  
4 repackaged traffic to a virtual protocol stack;  
5            sending said repackaged traffic to the intermediary layer;  
6            routing said repackaged traffic by the intermediary layer to an interface driver for  
7 a network interface having a node address memory;  
8            identifying a failed network interface having a node address; and  
9            storing the node address in the node address memory.

10  
11           20.    (Original)    A method for redundant networking in a network  
12 environment, comprising:

13           determining operative status of a first network interface having a first driver, and  
14 of a second network interface having a second driver with a driver memory for storing a  
15 MAC address for said second interface;

16           if the first network interface is inoperative, instructing the second driver to store  
17 the first network interface MAC address in the driver memory to allow processing by the  
18 second network interface of network traffic bound for the first network interface;

19           directing the second driver to activate the second network interface; and

20           directing the first driver to deactivate the first network interface.

1           21.   (Original)   A method according to claim 20, in which the network  
2 environment is a Novell based network, and wherein ODI commands are issued to said  
3 first and second drivers.

4  
5           22.   (Original)   A method according to claim 21, further comprising:  
6 receiving first network traffic by a protocol stack;  
7 forwarding said first network traffic to a LSL;  
8 routing said first network traffic from the LSL to a virtual MLID, and deriving  
9 second network traffic from said first network traffic;  
10 providing said second network traffic to a virtual protocol stack; and  
11 forwarding said second network traffic to the LSL.

12  
13           23.-25. (Canceled)

14  
15           26.   (Original)   A system, comprising:  
16 means for identifying a stored node address stored by a base driver for a network  
17 interface associated with the base driver; and  
18 means for directing the base driver to update the stored node address with a new  
19 node address.

20  
21           27.   (Original)   A system according to claim 26, further comprising:  
22 means for re-transmitting data, received in a first format from a network source,  
23 in a second format to a network destination; and



1 means for re-transmitting data, received in the second format from the network  
2 destination, in the first format to the network source.  
3

4 28. (New) An method for enhancing data network communication comprising:  
5 receiving network traffic for a network interface having a first node address;  
6 updating a stored node address stored in a receive address filtering table for a  
7 second network interface, and in a base driver for the second network interface, with the  
8 first node address; and  
9 routing the received network traffic to the second network interface.  
10

11 29. (New) The method of claim 28, wherein said receiving network traffic is  
12 performed by an intermediary configured to determine unavailability of the first network  
13 interface and automatically update the stored node address of the second network  
14 interface filtering table and its base driver so that the second network interface may  
15 transparently operate as if it were the first network interface.

---